UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/750,632	12/19/2003	Adam J. Simonoff	84734	9927

23501   7590   05/03/2007
NAVAL SURFACE WARFARE CENTER DAHLGREN DIVISION
OFFICE OF COUNSEL, CODE XDC1
17632 DAHLGREN ROAD
SUITE 121
DAHLGREN, VA 22448-5110

EXAMINER
KHATRI, ANIL

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
05/03/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

<table>
<tr><td rowspan="2">Office Action Summary</td><td>Application No.</td><td>Applicant(s)</td><td></td></tr>
<tr><td>10/750,632</td><td>SIMONOFF ET AL.</td><td></td></tr>
<tr><td></td><td>Examiner</td><td>Art Unit</td><td></td></tr>
<tr><td></td><td>Anil Khatri</td><td>2191</td><td></td></tr>
</table>

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

> A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
> - Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
> - If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
> - Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on <u>19 December 2003</u>.

2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) <u>1-38</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) <u>1-21, 23, 24, 26, 28, 29 and 33-38</u> is/are rejected.

7) ☒ Claim(s) <u>22, 25, 27 and 30-32</u> is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☒ The drawing(s) filed on <u>12/19/03</u> is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a) ☐ All   b) ☐ Some * c) ☐ None of:

      1. ☐ Certified copies of the priority documents have been received.

      2. ☐ Certified copies of the priority documents have been received in Application No. _____.

      3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date <u>12/19/03</u>.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____ .

DETAILED ACTION

Specification

The use of the trademark Java etc. has been noted in this application. It should be

capitalized wherever it appears and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary

nature of the marks should be respected and every effort made to prevent their use in any manner

which might adversely affect their validity as trademarks.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the

basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-21, 23, 24, 26, 28-29 and 33-38 are rejected under 35 U.S.C. 102(e) as being

anticipated by *Kodosky et al* USPN 7,200,838.

Regarding claims 1 and 34

Kodosky et al teaches,

providing asynchronous access to multiple users to a graphical programming and analysis

environment program (column 9, lines 14-24, graphical programming development

environment may provide an application programming interface (API) which is used in

programmatically generating the graphical program. For example, for each node, programmatic

structure, or other object of the graphical program, the API may be called to programmatically

add the object to the graphical program, connect the object to other objects of the graphical

program, etc. Thus, any necessary files or other constructs needed by the graphical

programming environment in order to use the generated graphical program may be

automatically created as a result of calling the API.

allowing each user to generate graphically represented code objects within the environment

program (column 7, lines 3-16, For example, when creating a state diagram using a state

diagram editor, the user may associate program code with one or more states in the state

diagram, e.g., by dragging and dropping program code icons from a palette onto the desired

states, or by using a dialog box to associate the program code with the states, etc. The program

code associated with a state may specify code to be executed when the state is active and/or

may specify transition condition code. The program code may comprise various types of

executable or source modules, such as a graphical sub-program, DLL or other shared library,

Java class, etc. The GPG program may include the program code in the generated graphical

program or may enable the generated graphical program to interface with the program code)

allowing each user access to the code objects of other users based on security privileges

accorded to the user (column 8, lines 10-24, the generated graphical program may be "locked",

requiring the user to explicitly unlock the graphical program before certain modifications to the

graphical program can be made within the graphical programming environment. Locking the

graphical program may ensure that the program's graphical source code is in a state known or

expected by the GPG program, thus facilitating the dynamic update of the graphical program.

For example, in one embodiment, the user may be allowed to add source code to the graphical

program that specifies execution instructions for each state and Boolean conditions for each

state transition, and may also be allowed to resize objects created by the GPG program, but may

be prevented from modifying the topology of the framework programmatically generated by the

graphical program until the graphical program is unlocked;

allowing each user to have the code objects of the user be chained to the code objects of other

users to which the user has access to yield inter-code object communication (column 11, lines

57-67, The GPG program may include a program (or program instructions) for creating the state

diagram information upon which the programmatically generated graphical program is based, or

the GPG program may be associated with or may interface with a separate program that creates

the state diagram information. For example, in one embodiment, the GPG program (or the

client portion of the GPG program) may include a state diagram editor program that allows a

user to interactively construct a state diagram. For example, such a state diagram editor may

allow the user to add and remove states, add and remove transition arrows between states,

define an initially active start state, define stop states, assign a priority ordering to transitions

exiting from a given state, etc. The state diagram editor may include a graphical user interface

that displays the current state of the state diagram as the diagram is constructed) and,

allowing each user to execute application programs made up of the code objects as chained

together within the environment program (column 11, lines 57-67, The GPG program may

include a program (or program instructions) for creating the state diagram information upon

which the programmatically generated graphical program is based, or the GPG program may be

associated with or may interface with a separate program that creates the state diagram

information. For example, in one embodiment, the GPG program (or the client portion of the

GPG program) may include a state diagram editor program that allows a user to interactively

construct a state diagram. For example, such a state diagram editor may allow the user to add

and remove states, add and remove transition arrows between states, define an initially active

start state, define stop states, assign a priority ordering to transitions exiting from a given state,

etc. The state diagram editor may include a graphical user interface that displays the current

state of the state diagram as the diagram is constructed).

Regarding claim 2

Kodosky et al teaches,

providing asynchronous access to the multiple users to the graphical programming and analysis

environment program comprises enabling multiple users to log into the environment program

remotely, such that the multiple users are able to access the environment program

simultaneously (column 7, lines 37-58, the GPG program may include, may be associated with,

or may be, a program used to construct the state diagram, e.g., a state diagram editor program.

For example, a user may utilize a state diagram editor to construct a desired state diagram and

may then request a graphical program based on the state diagram to be automatically created.

For example, the user may select a menu option to generate the graphical program. In response

to the user's request, the state diagram editor may execute program instructions to

programmatically generate the graphical program or may invoke a separate module or program

to perform the generation of the graphical program, passing the state diagram information (or a

reference to the information) to this separate module or program. In another embodiment, a

user may invoke the GPG program separately and request the GPG program to generate the

graphical program, e.g., by specifying a previously created state diagram file. The GPG

program may be operable to receive the state diagram information in any of various formats,

e.g., as binary data, XML data, text data, etc. The GPG program may process the state diagram

information to determine the specified states of the state diagram, the state transitions, etc.).

Regarding claims 3-10, 24 and 26

Rejection of claim 1 is incorporated and further claims 3-10 and 24 similar recites limitations as

in claim 1, therefore claims 3-10 and 24   are rejected under same rationale.

Regarding claims 11-12

Kodosky et al teaches,

the non-graphically represented code objects comprise one or more of: image-viewing

programs, video-playing programs, and audio-playing programs (column 7, lines 29-34, The

GPG program that generates the graphical program may be constructed using any of various

programming languages, methodologies, or techniques. For example, the GPG program may

itself be a graphical program, or the GPG program may be a text-based program, or the GPG

program may be constructed using a combination of graphical and text-based programming

environments).

Regarding claims 13-15 and 38

Kodosky et al teaches,

the graphical programming and analysis environment program is visually represented as a white

board (column13, lines 9-15, The one or more instruments may include one or more of a GPIB

instrument 112 and associated GPIB interface card 122, a data acquisition board 114 and

associated signal conditioning circuitry 124, a VXI instrument 116, a PXI instrument 118, a

video device 132 and associated image acquisition card 134, a motion control device 136 and

associated motion control interface card 138, and/or one or more computer based instrument

cards 142, among other types of devices).

Regarding claim 16

Rejection of claim 1 is incorporated and further claim recites limitation as in claim 2, therefore

claim 16 is rejected under same rationale.

Regarding claim 17

Kodosky et al teaches,

a plurality of graphically represented code objects, each code object created by a user and

accessible by other users in accordance with security privileges of the other users (column 8,

lines 10-24, the generated graphical program may be "locked", requiring the user to explicitly

unlock the graphical program before certain modifications to the graphical program can be

made within the graphical programming environment. Locking the graphical program may

ensure that the program's graphical source code is in a state known or expected by the GPG

program, thus facilitating the dynamic update of the graphical program. For example, in one

embodiment, the user may be allowed to add source code to the graphical program that specifies

execution instructions for each state and Boolean conditions for each state transition, and may

also be allowed to resize objects created by the GPG program, but may be prevented from

modifying the topology of the framework programmatically generated by the graphical program

until the graphical program is unlocked);

a plurality of graphically represented inter-code object connections, each inter-code object

connection representing data transfer between a pair of code objects ((column 7, lines 3-16, For

example, when creating a state diagram using a state diagram editor, the user may associate

program code with one or more states in the state diagram, e.g., by dragging and dropping

program code icons from a palette onto the desired states, or by using a dialog box to associate

the program code with the states, etc. The program code associated with a state may specify

code to be executed when the state is active and/or may specify transition condition code. The

program code may comprise various types of executable or source modules, such as a graphical

sub-program, DLL or other shared library, Java class, etc. The GPG program may include the

program code in the generated graphical program or may enable the generated graphical

program to interface with the program code)

one or more application programs made up of one or more chains of the code objects

interconnected via the inter-code object connections(column 11, lines 57-67, The GPG program

may include a program (or program instructions) for creating the state diagram information

upon which the programmatically generated graphical program is based, or the GPG program

may be associated with or may interface with a separate program that creates the state diagram

information. For example, in one embodiment, the GPG program (or the client portion of the

GPG program) may include a state diagram editor program that allows a user to interactively

construct a state diagram. For example, such a state diagram editor may allow the user to add

and remove states, add and remove transition arrows between states, define an initially active

start state, define stop states, assign a priority ordering to transitions exiting from a given state,

etc. The state diagram editor may include a graphical user interface that displays the current

state of the state diagram as the diagram is constructed).; and,

a graphical white board area within which the code objects are definable and movable and the

inter-code object connections are creatable (column 13, lines 30-36);

wherein the one or more application programs are executable within the graphical white board

area (column13, lines 9-15, The one or more instruments may include one or more of a GPIB

instrument 112 and associated GPIB interface card 122, a data acquisition board 114 and

associated signal conditioning circuitry 124, a VXI instrument 116, a PXI instrument 118, a

video device 132 and associated image acquisition card 134, a motion control device 136 and

associated motion control interface card 138, and/or one or more computer based instrument

cards 142, among other types of devices).

          Regarding claims 18-20

Kodosky et al teaches,

each code object is an applet program (column 7, lines 4-16, For example, when creating a state

diagram using a state diagram editor, the user may associate program code with one or more

states in the state diagram, e.g., by dragging and dropping program code icons from a palette

onto the desired states, or by using a dialog box to associate the program code with the states,

etc. The program code associated with a state may specify code to be executed when the state is

active and/or may specify transition condition code. The program code may comprise various

types of executable or source modules, such as a graphical sub-program, DLL or other shared

library, Java class, etc. The GPG program may include the program code in the generated

graphical program or may enable the generated graphical program to interface with the program code.

Regarding claims 21 and 36

Rejection of claim 17 is incorporated and further claim recites limitation as in claim 3, therefore claims 21 and 36 are rejected under same rationale.

Regarding claim 23

Kodosky et al teaches,

each inter-code object connection represents data being sent by a sender object of the pair of code objects and being received by a receiver object of the pair of code objects (column 8, lines 36-54, "GPG program...diagram file).

Regarding claim 28

Kodosky et al teaches,

One or more application programs are constructed one of asynchronously and synchronously (column 6, lines 27-34, "the state diagram... from the state).

Regarding claims 29 and 37

Kodosky et al teaches,

the one or more application programs are at least one of: capable of being stored for later retrieval and use, and modular in nature so that more complex application programs may be constructed therefrom (column 6, lines 55-67, "in one embodiment...state transition).

Regarding claim 33

Kodosky et al teaches,

a chat area within which the user can communicate with the other users ; and,

a user list area showing a name of each of the user and the other users currently logged into the

environment program (column 8, lines 25-41 and column 8, lines 61-67, In various

embodiments, the graphical program may be generated using any of various methods or

techniques. Generating the graphical program may comprise generating one or more files or

data structures defining the graphical program. When a user interactively develops a graphical

program from within a graphical programming environment, the graphical programming

environment may create one or more program files. For example, the program files may specify

information such as a set of nodes included in the graphical program, interconnections among

these nodes, programmatic structures such as loops, etc. In other cases, the program files may

store various data structures, e.g., in binary form, which the graphical programming

environment uses to directly represent the graphical program. Thus, in programmatically

generating the graphical program, the GPG program may programmatically generate one or

more files or data structures representing the graphical program, wherein these files may be

structured or formatted appropriately for a particular graphical programming environment.

Regarding claim 35

Kodosky et al teaches,

executing by the user of the application programs within the environment program (figure 6).

Allowable Subject Matter

Claims 22, 25, 27 and 30-32 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Anil Khatri whose telephone number is 571-272-3725. The examiner can normally be reached on M-F 8:30-5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ANIL KHATRI
PRIMARY EXAMINER